mn

Notice of Allowability	**Application No.** 10/695,970	**Applicant(s)** BEN-DAVID ET AL.	
	Examiner MARY STEELMAN	**Art Unit** 2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to 5/28/2007, 8/14/2007.

2. ☒ The allowed claim(s) is/are 1-33.

3. □ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
   a) □ All b) □ Some* c) □ None of the:
   1. □ Certified copies of the priority documents have been received.
   2. □ Certified copies of the priority documents have been received in Application No. _____ .
   3. □ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

4. □ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.

5. □ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
   (a) □ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
   1) □ hereto or 2) □ to Paper No./Mail Date _____.
   (b) □ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

   Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

6. □ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. □ Notice of References Cited (PTO-892)
2. □ Notice of Draftperson's Patent Drawing Review (PTO-948)
3. ☒ Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date 12/06/05
4. □ Examiner's Comment Regarding Requirement for Deposit of Biological Material
5. □ Notice of Informal Patent Application
6. □ Interview Summary (PTO-413), Paper No./Mail Date _____ .
7. ☒ Examiner's Amendment/Comment
8. ☒ Examiner's Statement of Reasons for Allowance
9. □ Other _____.

DETAILED ACTION

1. This Office Action is in response to Claim Amendments and Remarks received 05/28/2007. Claims 1-33 are pending.

Drawings

2. In view of the replacement sheet drawings for FIG. 1A, FIG. 1B, FIG. 2D, FIG. 5C, the prior objections are hereby withdrawn.

EXAMINER'S AMENDMENT

3. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Yoav Alkalay, Applicant's Representative and Suzanne Erez, Reg. No. 46,688 on 08/07/2007. (Proposed amendments faxed to Applicant on 08/13/2007. Acceptance response received 08/14/2007.)

The application has been amended as follows:

IN THE CLAIMS

1. (currently amended) A method for determining Single Instruction Multiple Data (SIMD) parallel execution configurations in a computer processor architecture, the method comprising:

identifying in a computer program [a] one or more loops that [is] are amenable to parallel execution on an SIMD processor;

identifying a memory access pattern of data required for implementing said
loop in said architecture;

computing a set of candidate configurations of resources required for processing said data in said architecture, wherein said computing [step] comprises configuring an indirect addressing vector pointer register of said architecture into a multiport, scalar register file containing independently addressable elements in support of either of reorder-on-read use and reorder-on-write use of a vector element file of said architecture;

selecting one of said candidate[s] configurations in accordance with predefined selection criteria; and implementing said [selected] SIMD parallel execution configuration in said architecture.

2. (currently amended) A method according to claim 1 where any of said [steps are] method is implemented by a compiler.

3. (currently amended) A method according to claim 1 wherein said computing [step] comprises configuring any of said vector pointer registers in support of loading a data vector into a plurality of non-contiguous segments of said vector element file of said architecture.

4. (currently amended) A method according to claim 1 wherein said computing [step] comprises configuring any of said vector pointer registers in support of loading a data vector into said vector element file of said architecture in support of a plurality of operations where each operation has a different access pattern.

5. (currently amended) A method according to claim 4 and further comprising:

performing any of said [steps] method for a plurality of said loops in the same computer program;

detecting a data reuse opportunity common to two or more of said loops; and

modifying any of said candidate configurations in support of said data reuse opportunity.

6. (currently amended) A method according to claim 1 and further comprising eliminating any of said candidate[s] configurations in accordance with predefined elimination criteria.

7. (currently amended) A method according to claim 6 wherein said eliminating [step] comprises eliminating any of said candidate[s] configurations that require[s] loading a data vector into said vector element file in a manner that cannot be accommodated by said vector element file.

8. (currently amended) A method according to claim 1 wherein said selecting [step] comprises selecting one of said candidate configurations that uses fewest vector pointer registers among all of said candidate[s] configurations.

9. (currently amended) A method for determining Single Instruction Multiple Data (SIMD) parallel execution configurations in a computer processor architecture for computations that feature arbitrary parametric access, the method comprising:

identifying in a computer program [a] one or more loops that accesse[s] data indirectly and that [is] are amenable to parallel execution on an SIMD processor;

determining that indices of said one or more loops fit within the range of a vector element file of said architecture, and, if so:

loading all loop data into said vector element file;

loading said indices into at least one indirect addressing vector pointer register of said architecture, said vector pointer register being configured into a multiport, scalar register file containing independently addressable elements in support of reorder-on-read use of said vector element file; and

processing said loop data.

10. (currently amended) A method according to claim 9 wherein said identifying [step] comprises identifying said one or more loops as performing a plurality of computations that operate in parallel on a permutation of data.

11. (currently amended) A method according to claim 9 where any of said [steps are] method is implemented by a compiler.

12. (currently amended) A system for determining Single Instruction Multiple Data (SIMD) parallel execution configurations in a computer processor architecture, the system comprising:

a processor with executable means for identifying in a computer program, a loop that is amenable to parallel execution on an SIMD processor;

a processor with executable means for identifying a memory access pattern of data required for implementing said loop in said architecture;

a processor with executable means for computing a set of candidate configurations of resources required for processing said data in said architecture, wherein said means for computing [step] is operative to configure [in a computer program a loop that is amenable to parallel execution on an

SIMD processor in support of either of reorder-on-read use and reorder-on-write use of a vector element file of said architecture] each of one or more indirect addressing vector pointer registers, of said architecture, into a multiport, scalar register file containing independently addressable elements, in support of either of reorder-on-read use and reorder-on-write use of a vector element file of said architecture;

a processor with executable means for selecting one of said candidate[s] configurations in accordance with predefined selection criteria; and

a processor with executable means for implementing said SIMD parallel execution [selected vectorization] configuration in said architecture.

13. (currently amended) A system according to claim 12 where any of said processor executable means are assembled with a compiler.

14. (currently amended) A system according to claim 12 wherein said processor executable means for computing is operative to configure any of said vector pointer registers in support of loading a data vector into a plurality of non-contiguous segments of said vector element file of said architecture.

15. (currently amended) A system according to claim 12 wherein said processor executable means for computing is operative to configure any of said vector pointer registers in support of loading a data vector into said vector element file of said architecture in support of a plurality of operations where each operation has a different access pattern.

16. (currently amended) A system according to claim 15 and further comprising:

processor executable means for [performing any of said steps] executing SIMD parallel execution configurations for a plurality of [said] loops in the same computer program;

processor executable means for detecting a data reuse opportunity common to two or more of said loops; and

processor executable means for modifying any of said candidate configurations in support of said data reuse opportunity.

17. (currently amended) A system according to claim 12 and further comprising processor executable means for eliminating any of said candidate[s] configurations in accordance with predefined elimination criteria.

18. (currently amended) A system according to claim 17 wherein said processor executable means for eliminating is operative to eliminate any of said candidate[s] configurations that requires loading a data vector into said vector element file in a manner that cannot be accommodated by said vector element file.

19. (currently amended) A system according to claim 12 wherein said processor executable means for selecting is operative to select one of said candidate configurations that uses the fewest vector pointer registers among all of said candidate[s] configurations.

20. (currently amended) A system for determining Single Instruction Multiple Data (SIMD) parallel execution configurations in a computer processor architecture for computations that feature arbitrary parametric access, the system comprising:

processor executable means for identifying in a computer program a loop that accesses data indirectly and that is amenable to parallel execution on an SIMD processor;

processor executable means for determining that indices of said loop fit within the range of a vector element file of said architecture, and, if so:

processor executable means for loading all loop data into said vector element file;

processor executable means for loading said indices into at least one indirect addressing vector pointer register of said architecture, said vector pointer register being configured into a multiport, scalar register file containing independently addressable elements in support of reorder-on-read use of said vector element file; and

processor executable means for processing said loop data.

21. (currently amended) A system according to claim 20 wherein said processor executable means for identifying is operative to identify said loop as performing a plurality of computations that operate in parallel on a permutation of data.

22. (currently amended) A system according to claim 20 where any of said processor executable means are assembled with a compiler.

23. (currently amended) A computer program embodied on a computer-readable medium, the computer program comprising:

a first code segment operative to identify in a computer program a loop that is amenable to parallel execution on an SIMD processor in a computer processor architecture;

a second code segment operative to identify a memory access pattern of data required [for] to implement said loop in said architecture;

a third code segment operative to compute a set of candidate configurations of resources required for processing said data in said architecture and configure an indirect addressing vector pointer register of said architecture into a multiport, scalar register file containing independently addressable elements in support of either of reorder-on-read use and reorder-on-write use of a vector element file of said architecture;

a fourth code segment operative to select one of said candidate[s] configurations in accordance with predefined selection criteria; and

a fifth code segment operative to implement said [selected vectorization configuration] SIMD parallel execution in said architecture.

24. (original) A computer program according to claim 23 where any of said code segments are assembled with a compiler.

25. (original) A computer program according to claim 23 wherein said third code segment is operative to configure any of said vector pointer registers in support of loading a data vector into a plurality of non-contiguous segments of said vector element file of said architecture.

26. (original) A computer program according to claim 23 wherein said third code segment is operative to configure any of said vector pointer registers in support of loading a data vector into said vector element file of said architecture in support of a plurality of operations where each operation has a different access pattern.

27. (currently amended) A computer program according to claim 26 and further comprising:

a sixth code segment operative to execute [perform any of said steps for] a plurality of [said] loops in the same computer program;

a seventh code segment operative to detect a data reuse opportunity common to two or more of said loops; and

an eighth code segment operative to modify any of said candidate configurations in support of said data reuse opportunity.

28. (currently amended) A computer program according to claim 23 and further comprising a ninth code segment operative to eliminate any of said candidate[s] configurations in accordance with predefined elimination criteria.

29. (currently amended) A computer program according to claim 28 wherein said ninth code segment is operative to eliminate any of said candidate[s] configurations that require[s] loading a data vector into said vector element file in a manner that cannot be accommodated by said vector element file.

30. (currently amended) A computer program according to claim 23 wherein said fourth code segment is operative to select one of said candidate configurations that uses fewest vector pointer registers among all of said candidate[s] configurations.

31. (currently amended) A computer program embodied on a computer-readable medium, the computer program comprising:

a first code segment operative to identify in a computer program a loop that accesses data indirectly and that is amenable to parallel execution on an SIMD processor in a computer processor architecture;

a second code segment operative to determine that indices of said loop fit within the range of a vector element file of said architecture, and, if so, load all loop data into said vector element file;

a third code segment operative to load said indices into at least one indirect addressing vector pointer register of said architecture, said vector pointer register being configured into a multiport, scalar register file containing independently addressable elements in support of reorder-on-read use of said vector element file; and

a fourth code segment operative to process said loop data.

32. (original) A computer program according to claim 31 wherein said first code segment is operative to identify said loop as performing a plurality of computations that operate in parallel on a permutation of data.

33. (original) A computer program according to claim 31 where any of said code segments are assembled with a compiler.

THE END

Allowable Subject Matter

4. Claims 1-33 are allowed.

5. The following is an examiner's statement of reasons for allowance:

Regarding independent claim 1 (and similarly in independent claims 9, 12, 20, 23, and 31), Ansari, Washio, and other cited prior arts, taken alone or in combination, fail to disclose:

"configuring an indirect addressing vector pointer register of said architecture into a multiport, scalar register file containing independently addressable elements in support of either of reorder-on-read use and reorder-on-write use of a vector element file of said architecture"

As noted on page 11 of Remarks, such an invention is limited to architectures that support SIMD parallel execution and the claimed invention uses an indirect addressing vector pointer register as detailed above.

Moreover, evidence for modifying the prior art teachings by one of ordinary skill level in the art was not uncovered so as to result in the invention.

Thus, remaining dependent claims, claims 2-8, 10, 11, 13-19, 21, 22, 24-30, 32, and 33, are allowed.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Conclusion

6. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mary Steelman, whose telephone number is (571) 272-3704. The examiner can normally be reached Monday through Thursday, from 7:00 AM to 5:30 PM If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached at (571) 272-3708. The fax phone number for the organization where this application or proceeding is assigned: 571-273-8300.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Mary Steelman

08/14/2007

MARY STEELMAN
PRIMARY EXAMINER

Mary Steelman